System Message Catalog

Thor DP2/DP3

Checkout and Launch Control Systems (CLCS)

84K00510-090

Approval:

Larry Wilhelm, Chief, Date
System Software Division

PREPARED BY: Jeff Wildgen, CLCS System Services

REVISION HISTORY

REV	DESCRIPTION	DATE

LIST OF EFFECTIVE PAGES				
Dates of issue of change pages are:				
Page No.	**A or D***	**Issue or Change No.**	**CR No.**	**Effective Date****

Table of Contents

SYSTEM MESSAGE CATALOG

THOR DP2/DP3

CHECKOUT AND LAUNCH CONTROL SYSTEMS (CLCS)

1. SYSTEM MESSAGE CATALOG (SMC) INTRODUCTION

1.1 SYSTEM MESSAGE CATALOG OVERVIEW

System Message Catalog is responsible for providing an environment where users can define and manage messages that are eventually displayed in the RTPS environment.

System Message Catalog provides a GUI interface that allows users to create and/or modify system messages, application messages and user application messages that will be stored in a SDC Message Catalog.

System Message Catalog also creates the SCID On-Line Message Catalog from the SDC SCID Message Catalog. As part of System Load, the SCID On-Line Message Catalog gets loaded on the CCWS(s) that will allow for easy and improved access of the message in the real-time RTPS environment. The SCID On-Line Message Catalog will contain both system and application messages.

System Message Catalog also creates the TCID On-Line Message Catalog(s) (based on RSYS) from the SDC TCID Message Catalog(s) for user application messages. The TCID On-Line Message Catalog(s) will become part of the TCID, and as part of the TCID Load, the TCID On-Line Message Catalog(s) get(s) loaded on the CCWS(s) as well.

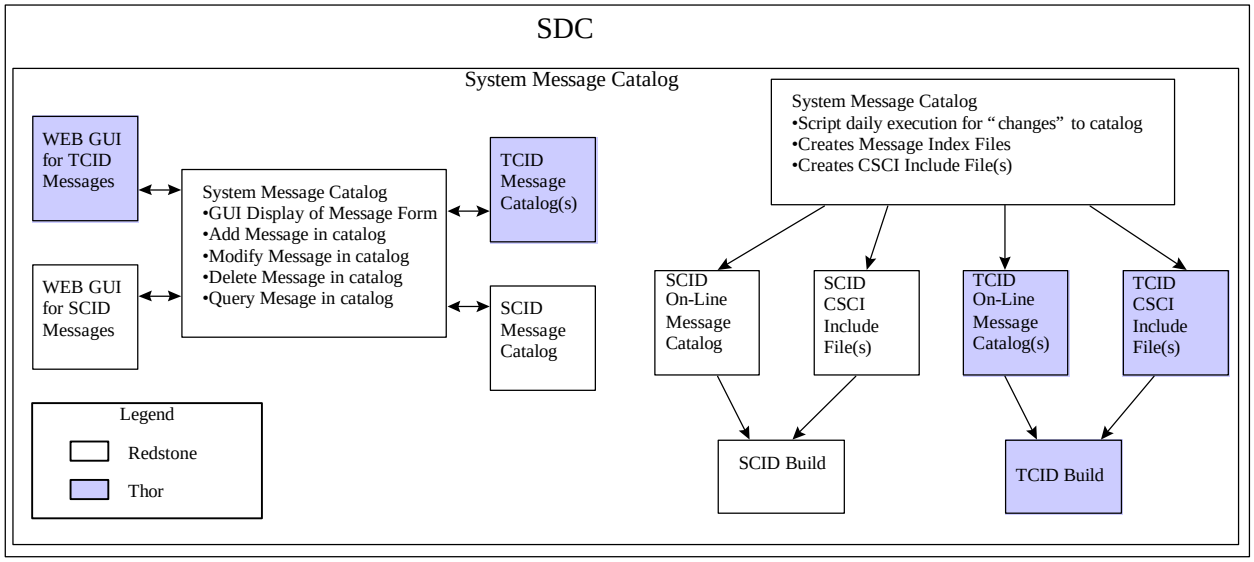

Figure 1. System Message Catalog Overview.

1.2 SYSTEM MESSAGE CATALOG OPERATIONAL DESCRIPTION

A Web Message Form GUI allows authorized database users to create/modify messages in the SCID and TCID Message Catalogs.

System Message Catalog provides a service as part of the SCID Build that creates the SCID On-Line Message Catalog from the SDC's SCID Message Catalog for the RTPS environment. As part of the TCID Build, SMC also creates the TCID On-Line Message Catalog(s) from the SDC's TCID Message Catalog(s).

As part of the SCID and TCID Build processes, the SMC automatic (daily) shell script invokes the SMC Create Index program, smc_create_index, that creates the SCID On-Line Message Catalog from the SCID DBM Message Catalog, smc_scid_messages.dbm, and the TCID On-Line Message Catalog(s) from the TCID DBM Message Catalog(s), smc_xxxx_messages.dbm (where xxxx corresponds to the RSYS).

The shell script automatically checks all On-Line Message Catalogs and DBM Message Catalog files into CM.

System Message Catalog also provides an API syntax checker against SMS APIs. This tool will be incorporated into CM and will verify that the API calls to SMS are correct in the source before it is checked in to CM.

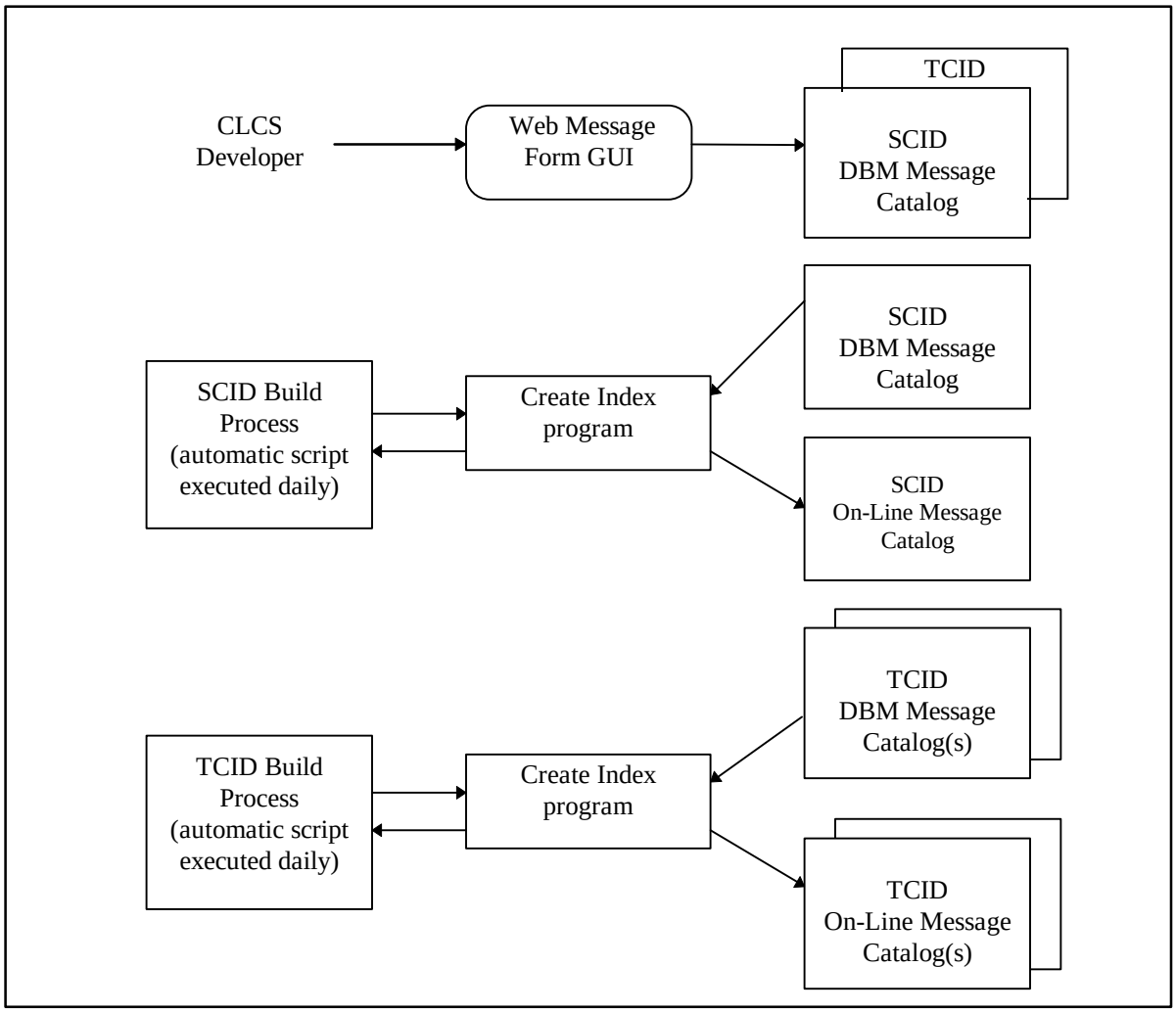

Figure 2. System Message Catalog High Level Operational Description.

1.2.1 Conversion Process of the Message Database Catalog

- The SMC Create Index program creates the SCID and TCID On-Line Message Catalogs from the SCID and TCID Message Catalogs.
- The SMC Create Index program creates the respective CSCI Include File(s).
- The CSCI Include File(s) are utilized by the various CSCIs that issue their respective Message(s).
- The SMC Create Index program removes any trailing CR/LF (s) from the message text for the System Message Viewer.
- The SCID On-Line Message Catalog becomes part of the SCID, loaded on the RTPS on the various CCWS(s).
- The TCID On-Line Message Catalog(s) become(s) part of the TCID, loaded on the RTPS on the various CCWS(s).

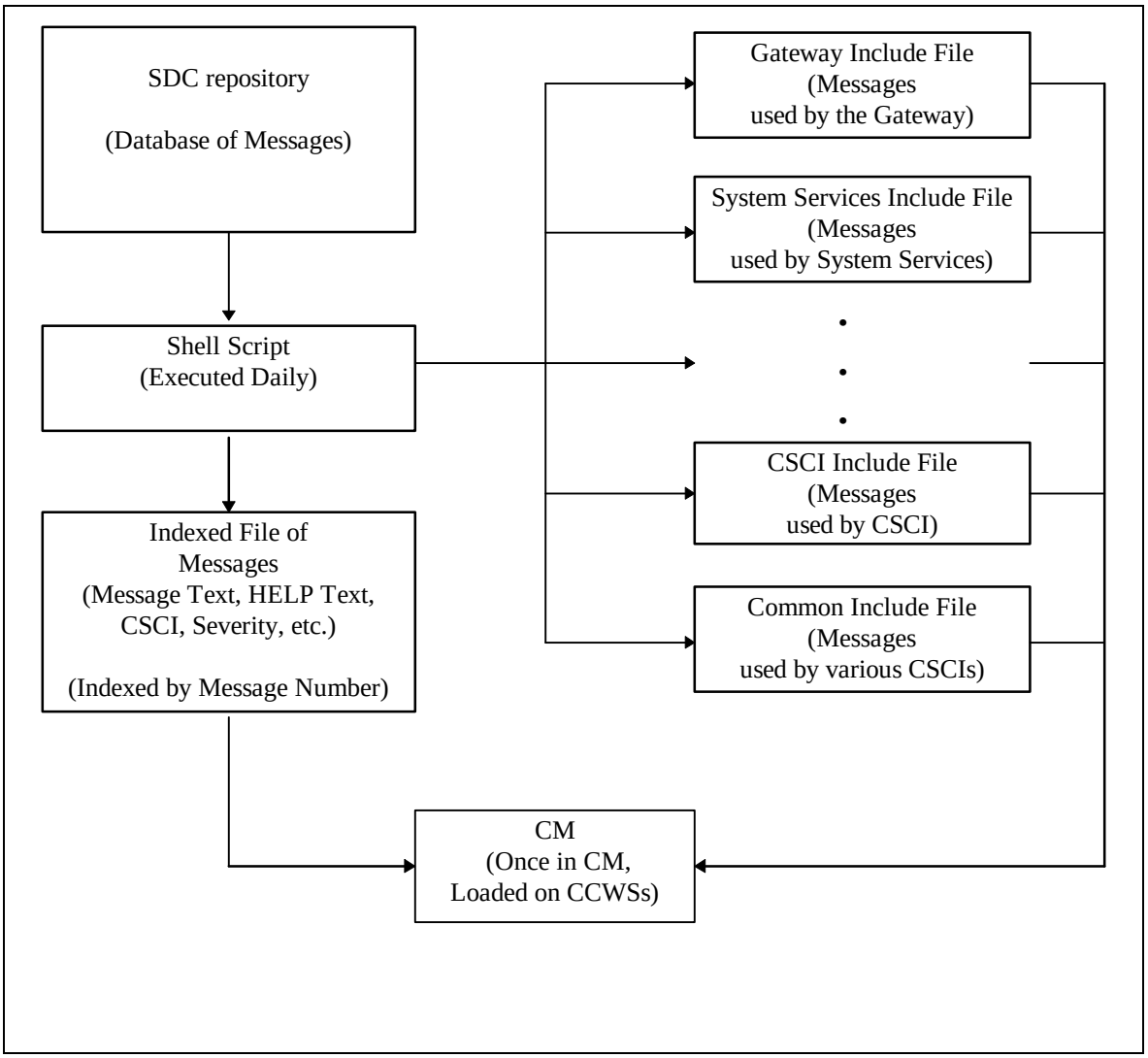

Figure 3. Conversion Process Operational Description

2. SYSTEM MESSAGE CATALOG SPECIFICATIONS

2.1 SYSTEM MESSAGE CATALOG GROUND RULES

Ground Rules that are post-Thor are in italics.

- CM SCID Build will execute the System Message Catalog conversion program that creates the SCID On-Line Message Catalog and CSCI Include files for System and Application Messages.
- CM TCID Build will execute the System Message Catalog conversion program that creates the TCID On-Line Message Catalog(s) and CSCI (i.e., RSYS) Include files for User Application Messages.
- CM SCID Build will provide a command-line macro to submit the CSCI Include files into CM.
- CM SCID Build will prevent the CSCI Include file from being checked into CM if the file has not changed.
- Users will only be allowed to fetch (not checkout for update) the CSCI Include files from CM.
- CM SCID Build must include the SCID On-Line Message Catalog as part of the SCID used by the RTPS environment.
- CM SCID Load must include the SCID On-Line Message Catalog on every CCWS that provides a System Message Viewer.
- CM TCID Build must include the TCID On-Line Message Catalog as part of the TCID used by the RTPS environment.
- CM TCID Load must include the TCID On-Line Message Catalog on every CCWS that provides a System Message Viewer.
- *CM must invoke the API Syntax Checker upon source check-in.*

2.2 SYSTEM MESSAGE CATALOG FUNCTIONAL REQUIREMENTS

Requirements that are post-Thor are in italics.

2.2.1 System Message Catalog GUI Form Requirements

1. SMC will create a GUI to submit messages into the Message Catalog
2. A Message Catalog will reside on the SDC as a database.
3. The Message Form will allow the user to create/modify/delete/query SCID messages in the Message Catalog.
4. The Message Form will allow the user to create/modify/delete/query TCID messages in the Message Catalog.
5. The Message Form will provide on-line help.
6. The Message Form will support a minimum of 3 message severity level classifications: Informational, Warning, and Error.
7. SMC will provide the capability to specify Message Type in all messages (i.e. “Summary”, “Intermediate”, “Details”, or “Other”).

8. The Message Form will allow a maximum of 30 message parameters (i.e., message inserts) to be specified.
9. The Message Form will require HELP text to be entered by the submitter for each Message created.
10. The Message Form will allow a maximum of 200 characters to be specified for the Message Text.
11. The Message Form will allow a maximum of 5 lines to be specified for the Message Text.
12. The Message Form will allow a maximum of 4000 characters to be specified for the Message Help Text.
13. SMC will provide a mechanism to associate an audible alarm with a message.
14. The Message Form will verify that the insert types in the Message Text are valid (e.g., %s, %d, %x, %o, etc.).
15. The Message Form will provide password protection whenever messages need to be created, modified, or deleted.
16. The Message Form will provide the capability for users to provide Insert Details to Message Inserts.
17. The Message Form will allow a maximum of 2000 characters to be specified for the Insert Details.
18. The Message Form will reuse message numbers from previously deleted messages.
19. The Message Form will provide the capability of non-positional message inserts in the message text of the GUI message submission (e.g., %1d, %2s, %3x etc.). This will provide the users the capability to reword the text and reorder the insert fields (e.g., %d, %s, %x, etc.) without requiring a recompile of the software.
20. The Message Form will validate the fields for valid input (Message Mnemonic and Insert Types).
21. The Message Form will provide the capability to specify a percent (%) symbol in the Message Text.
22. The repository mechanism (i.e., the Message Catalog) will be the same for System, Application and User Application Messages as well.
23. SMC will provide the capability to specify the RSYS in all messages. The RSYS will default to “SMG” for SCID messages.
24. SMC will provide the capability to share commonly used messages through the use of a “CMN” CSCI category for SCID messages.
25. *SMC will provide a softcopy version (HTML-based) of both System Message Catalogs on the various CCWSs that are Web-Enabled.*
26. *The Message Form will provide users the ability to query the HTML-based System Message Catalogs on their Web-Enabled workstation [TBD - CCWS and/or SWS].*

2.2.2 System Message Catalog Conversion Program Requirements

1. SMC will place the Message Catalog under CM control.
2. SMC will support CM multiflow for the Message Catalog and On-Line Message Catalogs.
3. The SMC Conversion program will create the SCID On-Line Message Catalog (indexed by message number) based on the SCID Message Catalog contents for System and Application Messages.

4. The SMC Conversion program will create the TCID On-Line Message Catalog(s) (indexed by message number within the catalog, catalog is determined by RSYS) based on the TCID Message Catalog(s) contents for User Application Messages.
5. The SMC Conversion program will create CSCI include files that contain the MNEMONIC and the corresponding message number.
6. The SMC Conversion program will follow the following naming convention for the include files: ***xxx_msgs.h*** where xxx is the CSCI name (e.g., ***cmd_msgs.h*** for the CMD CSCI).
7. The SMC Conversion program will remove any trailing CR/LF (s) from the message text.
8. The SMC Conversion program will create CSCI (i.e., based on RSYS) include files that contain the MNEMONIC and the corresponding message number for User Application Messages.
9. The SMC Conversion program will place the CSCI Include files under CM control upon creation/modification of the CSCI Include file(s) under the **SCIDBLD** userid.
10. *For the TCID Build, The SMC Conversion program will accept as an input parameter, a file name that will consist of a list of applicable RSYS's.*

2.2.3 System Message Catalog Syntax Checking Requirements

1. *SMC will provide an API Syntax Checker tool (as part of TCID and SCID Build) that will verify the correctness of the software's System Message API calls to the respective System Message in the Message Catalog.*
2. *The API Syntax Checker will validate that the Message Number/Mnemonic from the API call exist in the Message Catalog.*
3. *The API Syntax Checker will validate that the Number of Inserts specified in the API call matches the Number of Inserts defined in the Message Catalog.*
4. *The API Syntax Checker will validate that the Number of Inserts in the API call matches the total insert arguments listed in the API call.*
5. *The API Syntax Checker will validate that the Insert Types specified in the API call match the insert types defined in the Message Catalog. This function will support Non-Positional Inserts.*
6. *The API Syntax Checker will examine the source code for impacts of Deleted and Modified Messages.*

2.3 SYSTEM MESSAGE CATALOG PERFORMANCE REQUIREMENTS

1. None identified for Thor.

2.4 SYSTEM MESSAGE CATALOG INTERFACES DATA FLOW DIAGRAMS

This section provides a description and diagram of all of the interfaces to System Message Catalog .

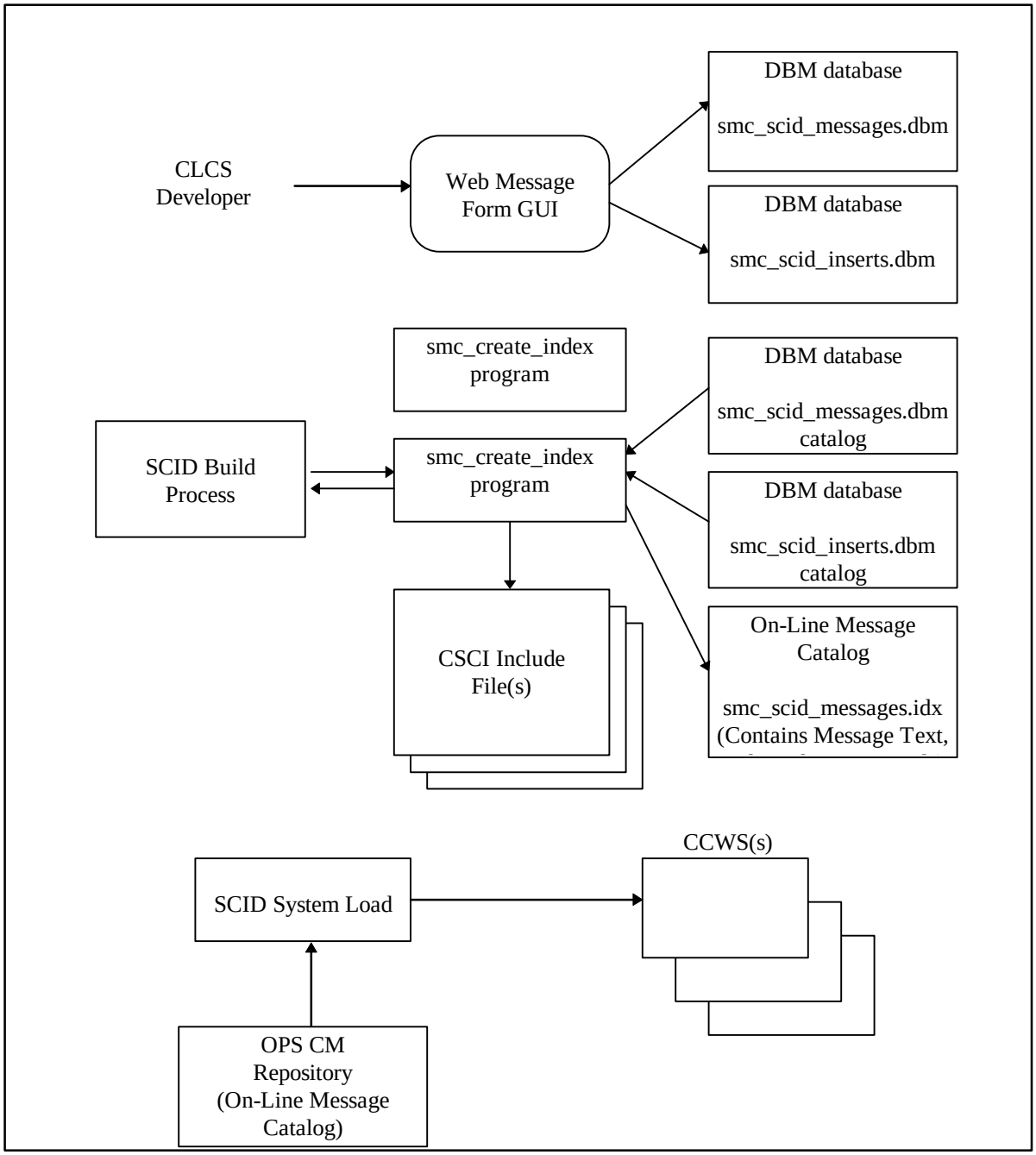

Figure 4. SCID Data Flow Diagram

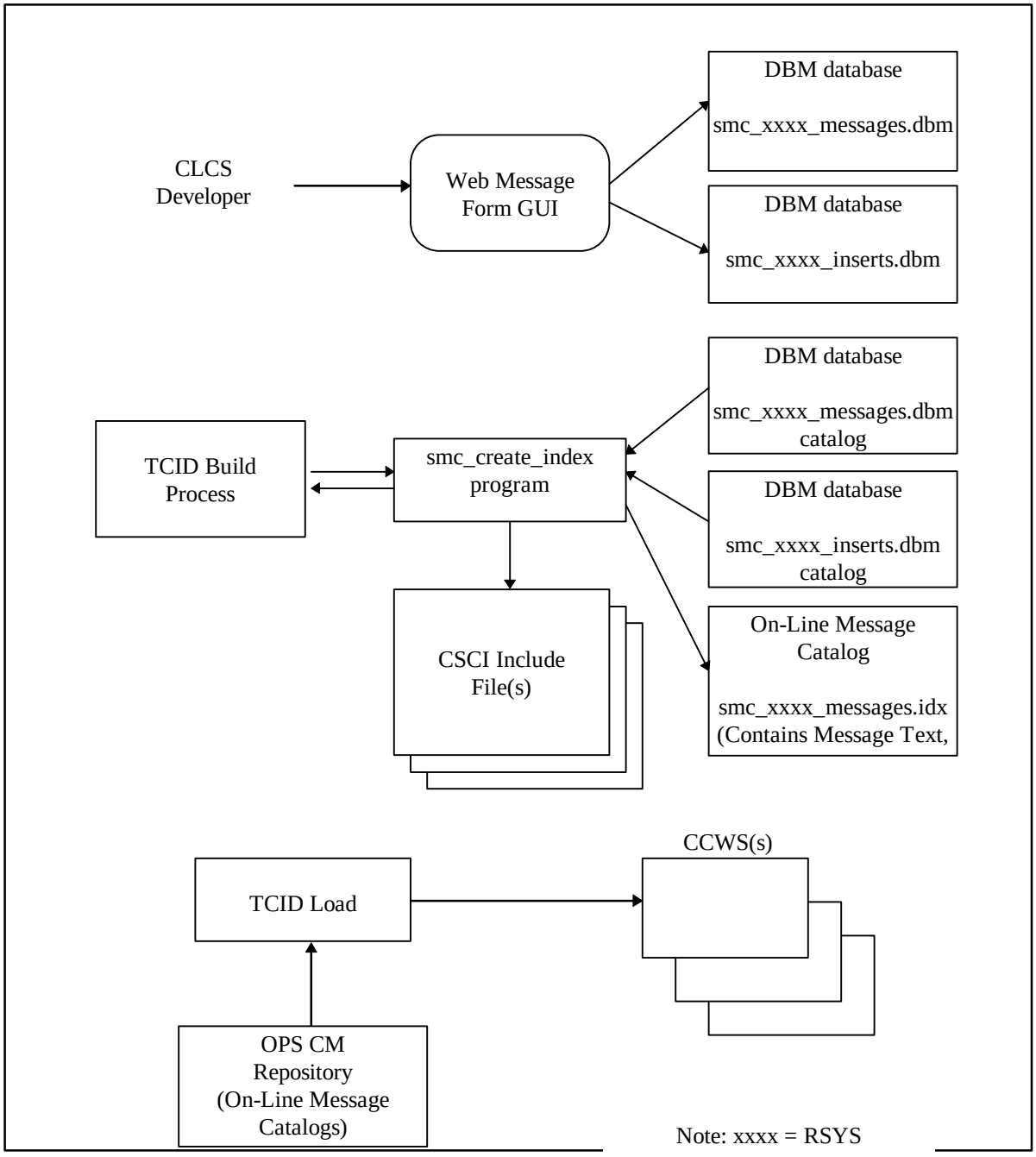

Figure 5. TCID Data Flow Diagram

3. SYSTEM MESSAGE CATALOG DESIGN SPECIFICATION

3.1 SYSTEM MESSAGE CATALOG DETAILED DATA FLOW

This data flow provides a pictorial representation of the data flow between external sources and destinations and the major and minor functions of System Message Catalog.

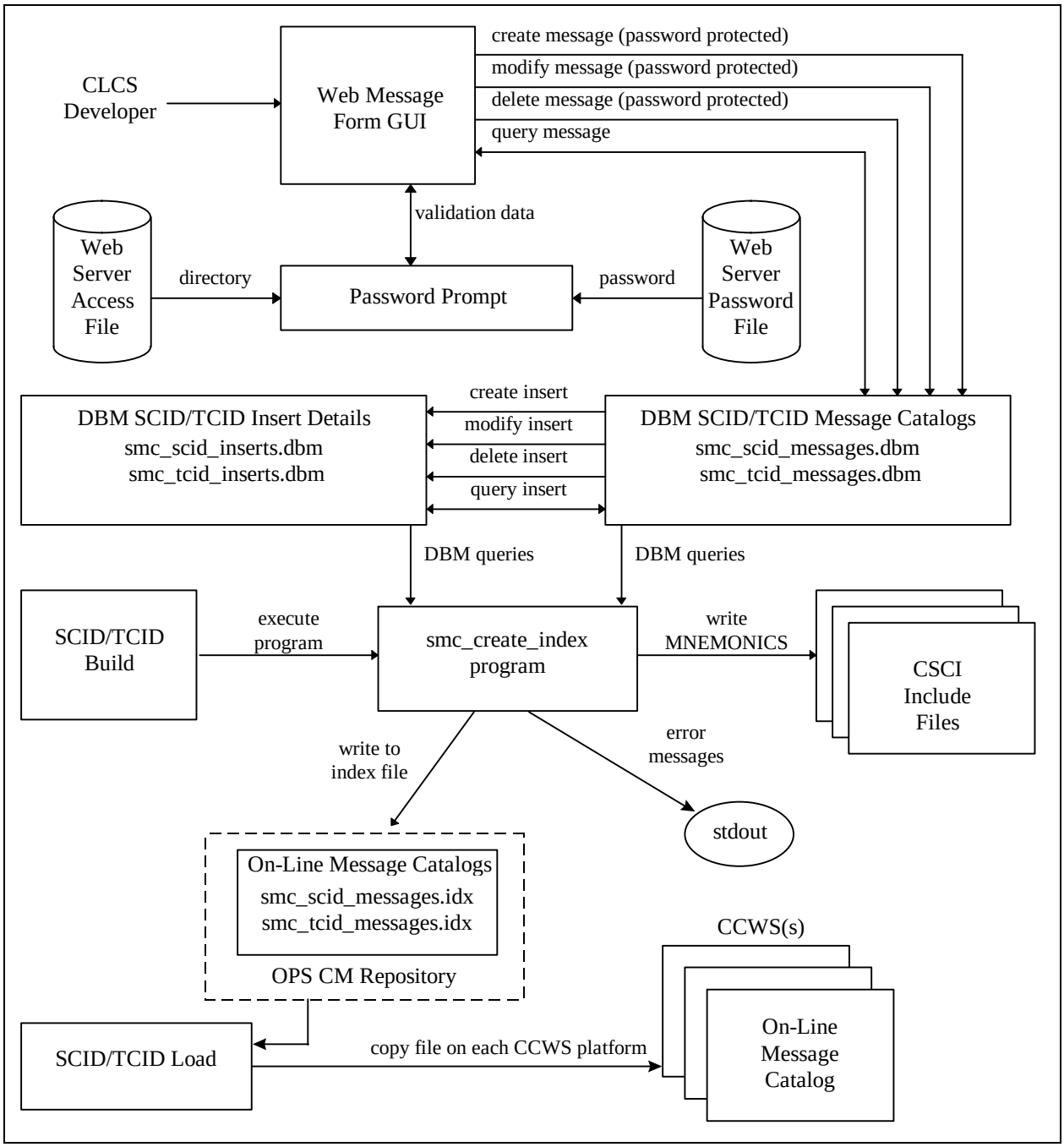

Figure 6. Detailed Data Flow Diagram

3.2 SYSTEM MESSAGE CATALOG EXTERNAL INTERFACES

3.2.1 System Message Catalog Message Formats

SMC generates the following messages during the execution of the Message Create Index program. The messages are preceded with the respective function prefix (*MsgIndexBuilder, MsgIncludeFileBuilder*). All messages are directed to stdout.

Messages generated by Message Index Builder
"Entry exceeds three lines in height: %d"
Inserts: Message number

Messages generated by Message Index Builder and Message Include File Builder
"Out of memory (fatal)"
Inserts: None

"Fatal error opening file %s"
Inserts: filename

"Fatal error writing to file %s"
Inserts: filename

"Fatal error reading from file %s"
Inserts: filename

"Error closing file %s"
Inserts: filename

"Attempt to call member function after a fatal error occurred: %s"
Inserts: Name of member function

Messages generated by Message Include File Builder
"(%s): Attempt to write to an unopened file %s"
Inserts: Current filename, file to which a write was attempted

"(%s): Non-fatal error opening file %s"
Inserts: Current filename, file the class tried to open

"(%s): Entry includes an invalid character and may not compile: %d, '%c' (%s)"
Inserts: Current filename, message number, invalid character, mnemonic text

3.2.2 System Message Catalog Display Formats

The following pictorials represent the GUI Message form used to create/modify/delete/query messages in the Message Catalog. All SMC web pages have a consistent look and feel. A toolbar is provided with functions common to all pages: Back, Help, Reset, and Find. The Search Capability is accessible from any page and can search either Message Catalog by a particular field

3.2.2.1 GUI Display: CSCI Selection

The CSCI Selection form displays a listing of both SCID and TCID CSCIs. Selection of one of the CSCIs brings up the Password Protection box for authentication to edit or create messages in the chosen CSCI. Reference *Figure 7* for an example of the CSCI Selection form.

3.2.2.2 GUI Display: Password Protection

Password Protection is achieved by making use of the web server's built in file and directory locking features via the *.htaccess* file located in the SMC main directory. Users are restricted to their own CSC through the use of a common CSC password. Passwords are stored in the password file *.smc-passwd* in the directory pointed to by the *.htaccess* file. Passwords may be changed by users who have write access to both the *.htaccess* and *.smc-passwd* files through the use of the web server utility *htpasswd*. Reference *Figure 8* for an example of the Password Prompt.

3.2.2.3 GUI Display: Message Submission

The Message Submission form provides the interface to edit and create messages in the respective SCID or TCID Message Catalog. It is accessible only through the CSCI Selection form, and will prime the drop-down boxes with selections relevant to the current CSCI. Users must authenticate through the Password Protection box in order to be able to edit messages here. Reference *Figure* 9 and *Figure 10* for an example of the Message Submission form.

3.2.2.4 GUI Display: Insert Details

Once the user has submitted the message, the Insert Details form will be displayed, allowing the user to enter detail information associated with each insert. Reference *Figure 11* for an example of the Insert Details form.

3.2.2.5 GUI Display: Message Validation

Validation is performed on the submitted message before it is entered into the Message Catalog. Inserts are checked for validity and count and the Mnemonic is checked for C-

Compilable characters. If any validation fails, the user is prompted with a description of the error and suggestion on how to correct.

3.2.2.6 GUI Display: Search Capability

The Search Capability may be executed from any web page. If it is executed from the Message Submission page, the form will be primed with any search criteria specified on the Message Submission page. This is done to maintain use compatibility with previous versions of the Message Submission page. There is also the capability to provide a sort preference for the find results. This is a pull-down list of text fields in the Message Catalog that the user may choose to sort by. If no sort is specified, the default is the Message Mnemonic field. If the search found any messages, a form opens allowing the user to edit or delete any of the found messages. Password Protection is enforced if the user has not already done so. Reference *Figure 15* for an example of the Search form.

3.2.2.7 GUI Display: Help Form

The Help Form is accessible from any SMC form. Each form defaults the Help Form to the specific information about the current form. A new web browser for Help will be opened if it does not already exist to avoid losing any unsubmitted changes to the current form. Reference *Figure 12*, *Figure 13*, and *Figure 14* for examples of the Help Form.

The GUI forms are captured on the following pages.

Note: The following screen capture pages do not reflect the latest changes.

Figure 7. Web Message Form GUI display - *CSCI Selection*

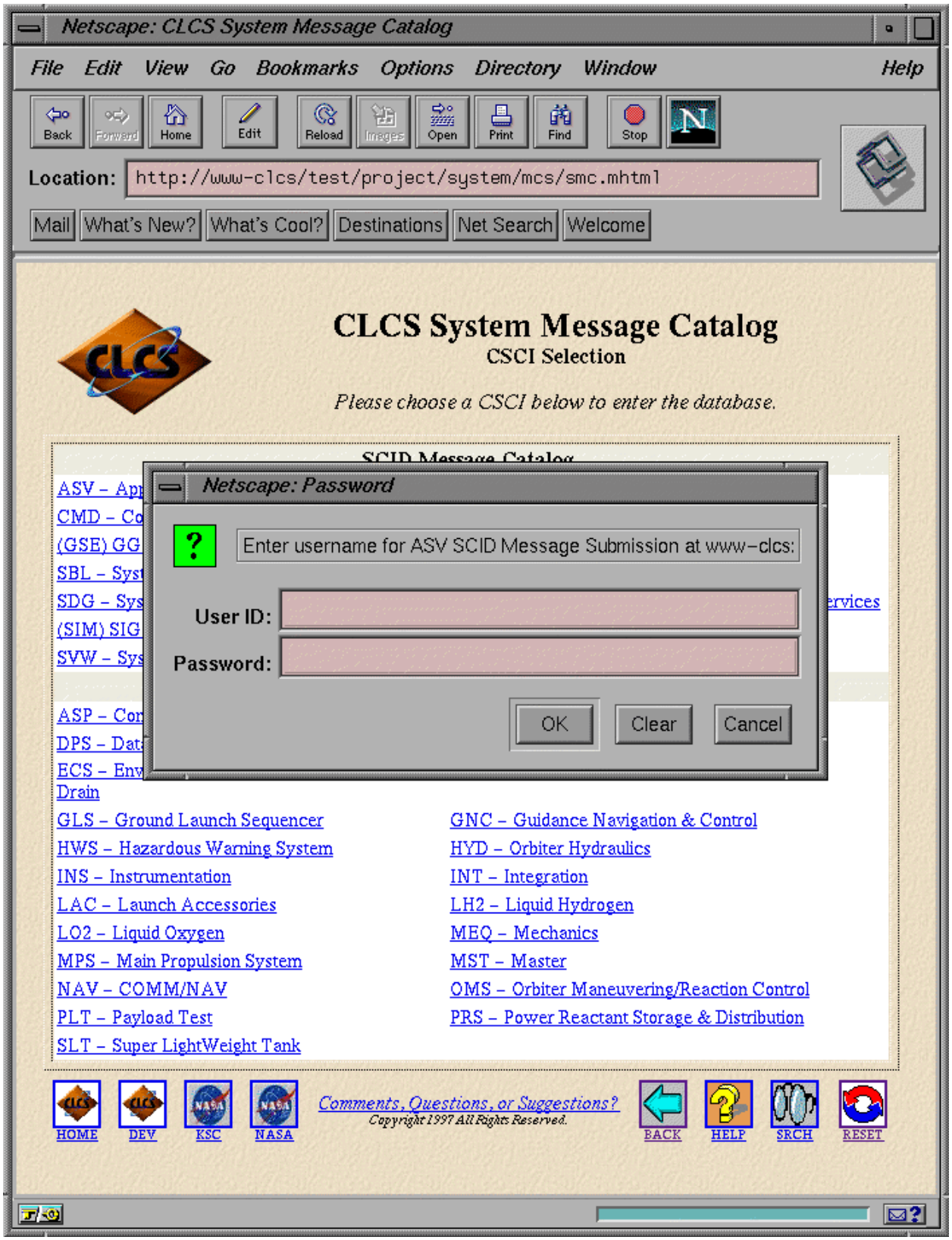

Figure 8. Web Message Form GUI display - *Password Validation*

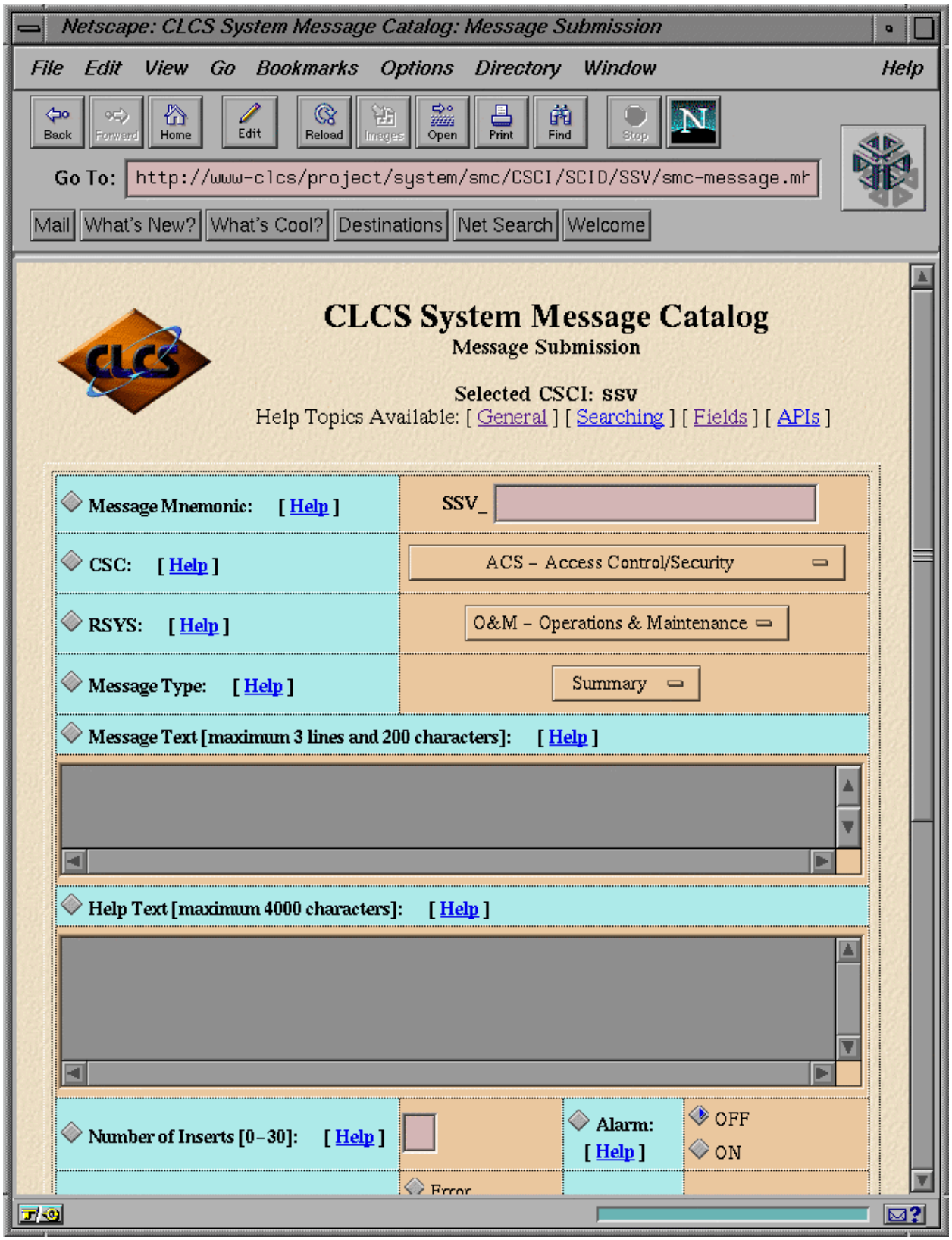

Figure 9. Web Message Form GUI display - *Message Submission*

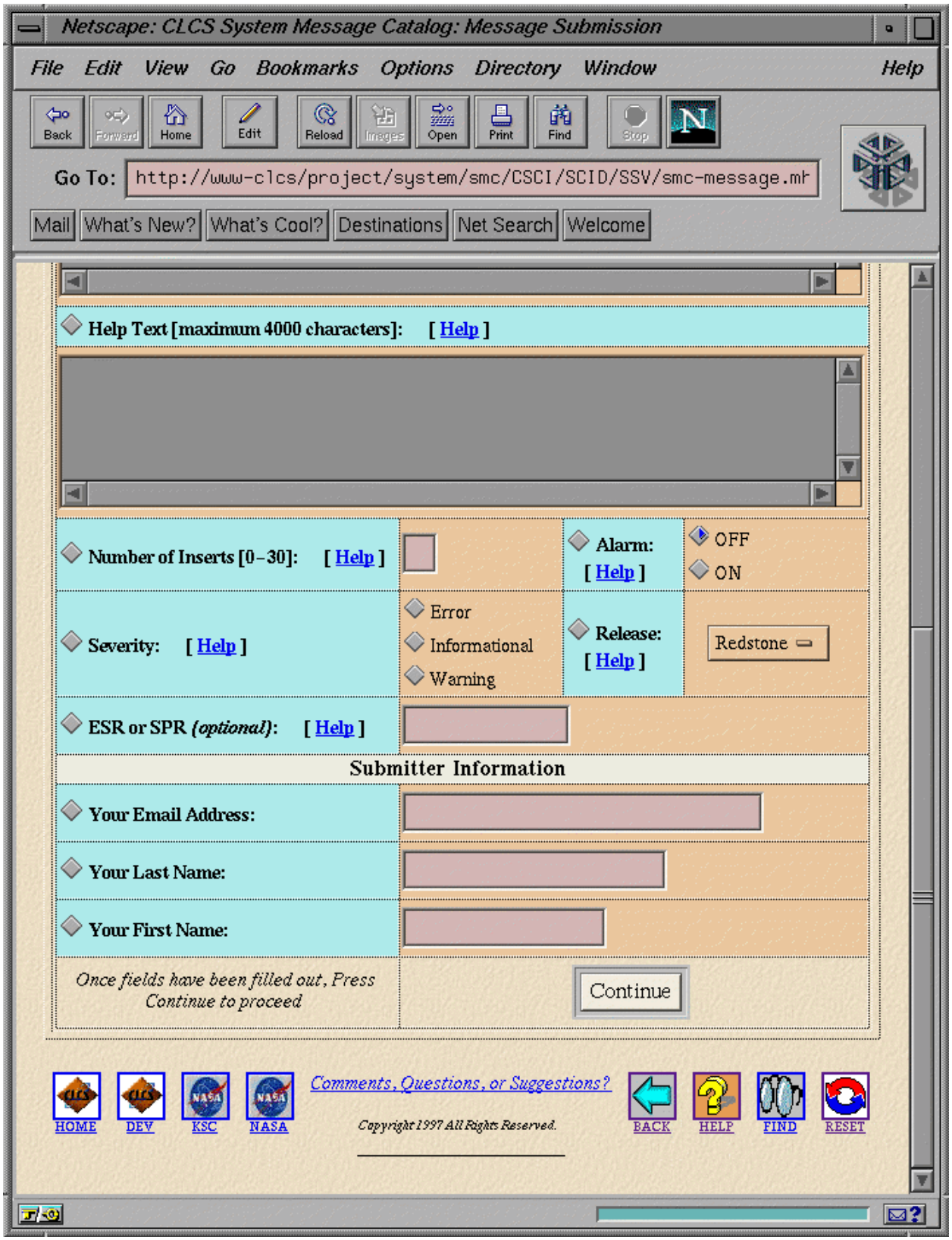

Figure 10. Web Message Form GUI display - *Message Submission* - Rest of Web page.

Figure 11. Web Message Form GUI display - *Insert Details*

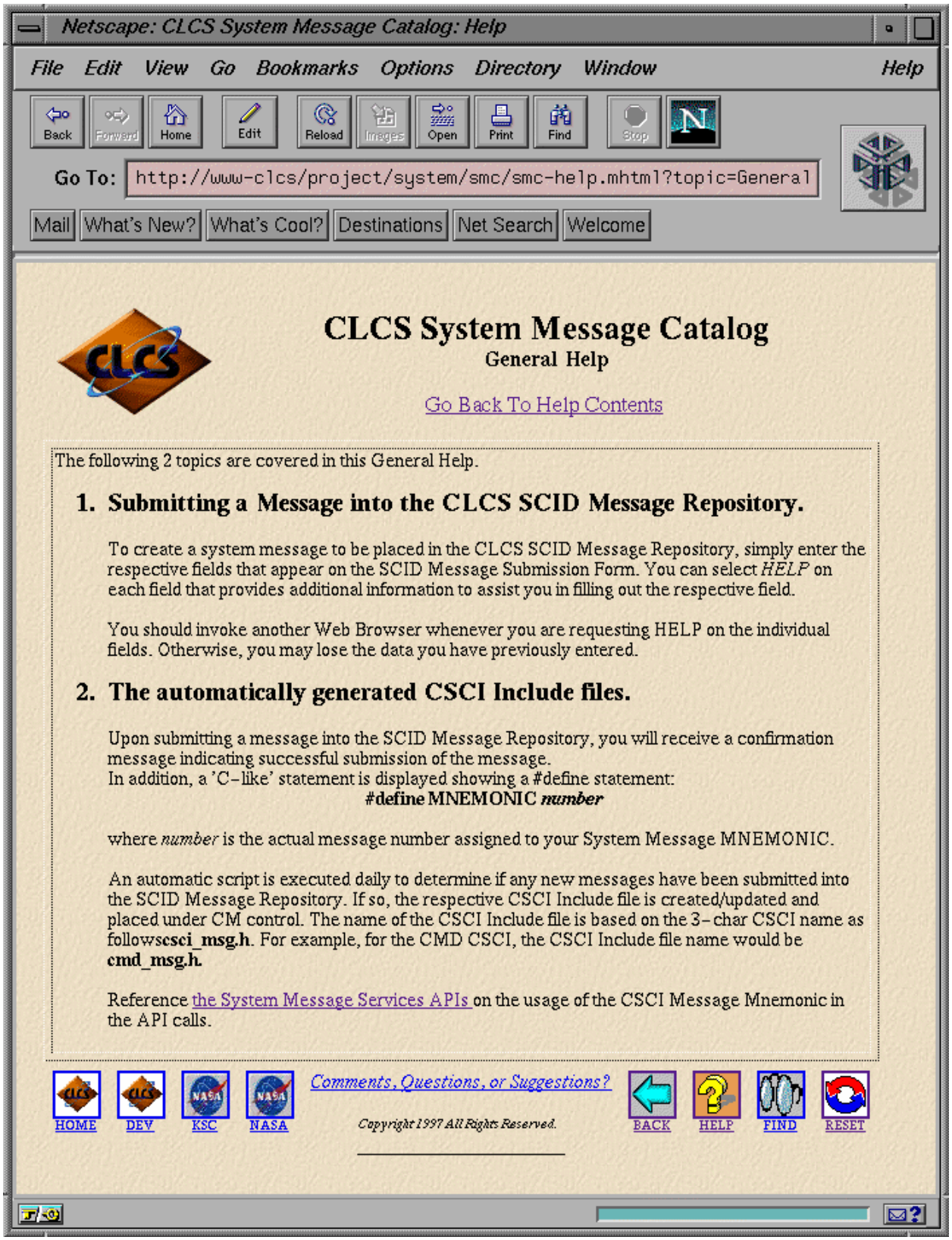

Figure 12. Web Message Form - *Help [General]*

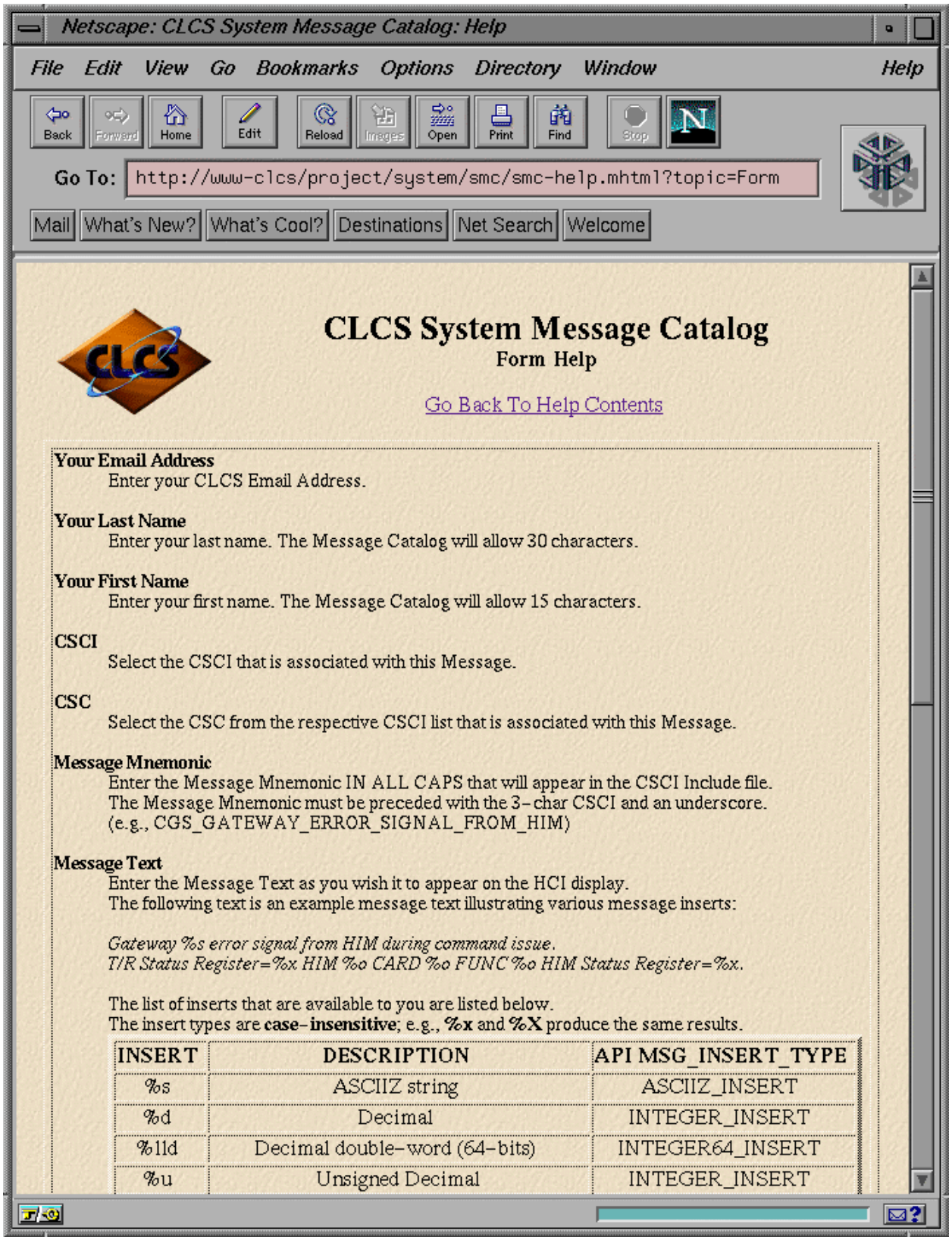

Figure 13. Web Message Form - *Help [Fields]*

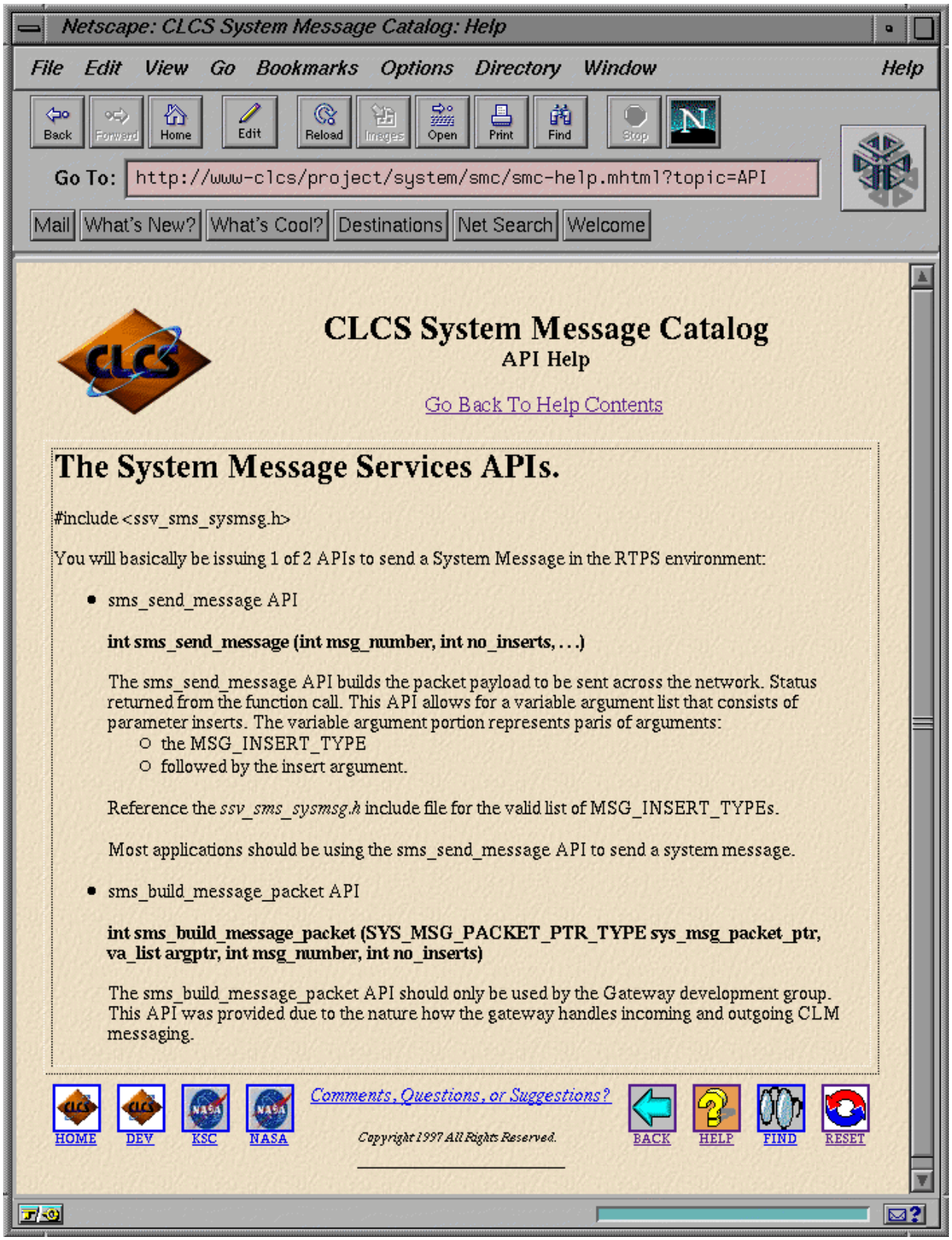

Figure 14. Web Message Form - *Help [APIs]*

Figure 15. Web Message Form - *Search*

3.2.3 System Message Catalog Input Formats

The table below shows the various Insert Types that can be declared in the Message Text field of the web Message Submission form.

CCMS System Msg Insert Type	***CLCS Insert Declaration*** *(Utilizing sprintf type)*	***Comments***
ASCII	%s	string is null-terminated
Decimal	%d	16-bit integer
Double-Precision Decimal	%d	32-bit integer
	%u	unsigned decimal
Floating Point	%f	[-]mmm.nnnnn notation
(new)	%e	[-]m.nnnnnne[+]xx notation
HEX	%x	uppercase HEX representation
OCTAL	%o	Octal representation
Binary	%b	Binary representation
CDT	%cdt	ddd:hhmm/ss notation
GMT	%gmt	hhmm/ss.sss notation
(new)	%mid	Message ID, uppercase HEX
Double-Word (64-bits) *(Utilizing sprintf syntax where appropriate)*		
(new)	%lld	Decimal double-word
	%llu	"
(new)	%lle	Float double-word
	%llf	"
	%llg	"
(new)	%llx	uppercase Hex double-word
(new)	%llo	Octal double-word
(new)	%llb	Binary double-word

Figure 16. Message Insert Type Definitions

3.2.3.1 Message Insert Formatting scheme

- When the message is formatted by System Message Services (SMS), SMS utilizes the "C" language sprintf function when formatting the Message Text Inserts. However, the following insert types, %b, %llb, %cdt, %gmt, and %mid are unique to CLCS and therefore are handled as special cases by the SMS formatting function.
- The Message Inserts are case-insensitive. In other words, %s and %S have the same effect.
- To specify a literal percent symbol, type two percent symbols together ("%%").
- Non-Positional Inserts are specified by inserting the position number between the percent symbol and the Insert Type (e.g. "%1s" or "%2x").
- Note, the sprintf lowercase %x is supported as a %X - the HEX representation in UPPERCASE.
- Reference the System Message Services design package for additional details.

3.2.3.2 Message Catalog Entry Example

Below is an example illustrating the various insert types. (Reference *Appendix A* for the CCMS System Message Catalog layout for System Message 197.)

1. **Message 197 as it would appear in the SDC Message Catalog database and the RTPS On-Line Message Catalog:**

   Gateway %s error signal from HIM during command issue.
   T/R Status Register = %x HIM %o CARD %o FUNC %o
   HIM Status Register = %x.

2. **The SMS_Send_Message API (issued by an application on the RTPS network):**

   Note: The application will most likely pass variables and not the actual values - this is for illustration purposes only.

```
sms_send_message (
      CGS_ERROR_SIGNAL_FROM_HIM, 6,
      ASCIIZ_INSERT, “GS1A”,
      INTEGER_INSERT, 0x5A63,
      INTEGER_INSERT, 07,
      INTEGER_INSERT, 03,
      INTEGER_INSERT, 06,
      INTEGER_INSERT, 0x6B5F)
```

   Where 6 = number of inserts
   ASCIIZ_INSERT is the type of insert that follows
   “GS1A” is the ASCIIZ insert for the failed Gateway
   INTEGER_INSERT is the type of insert that follows
   “0x5A63” is the HEX number of the Transmitter/Receiver Status Register Contents
   “07” is the OCTAL number of the HIM Number
   “03” is the OCTAL number of the CARD Number
   “06” is the OCTAL number of the FUNCTION CODE of HIM
   “0x6B5F” is the HEX number of the HIM Status Register

   Note: The above API call was for illustration purposes only. It does not require the application to pass the integer insert parameters in the necessary format, but the integers can be passed in decimal:

```
sms_send_message (
      CGS_ERROR_SIGNAL_FROM_HIM, 6,
      ASCIIZ_INSERT, “GS1A”,
      INTEGER_INSERT, 23139,
      INTEGER_INSERT, 7,
      INTEGER_INSERT, 3,
      INTEGER_INSERT, 6,
      INTEGER_INSERT, 27487)
```

   Reference System Message Services package for additional information about the sms_send_message API..

3. **The formatted message will appear as follows when displayed:**
   Reference System Message Viewer for prepended information such as Timestamp & CSCI.

   Gateway GS1A error signal from HIM during command issue.
   T/R Status Register = 5A63 HIM 7 CARD 3 FUNC 6
   HIM Status Register = 6B5F.

4. **Message 197 Correctly Utilizing Non-Positional Inserts:**
   Gateway %1s error signal from HIM during command issue.
   T/R Status Register = %2x HIM %3o CARD %4o FUNC %5o
   HIM Status Register = %6x.
   or…
   Gateway %1s error signal from HIM during command issue.
   HIM Status Register = %6x HIM %3o CARD %4o FUNC %5o
   T/R Status Register = %2x.

   Note: This would allow the message to be reworded in the Message Catalog without forcing a recompile. The API call would remain the same.

```
sms_send_message (
      CGS_ERROR_SIGNAL_FROM_HIM, 6,
      ASCIIZ_INSERT, "GS1A",
      INTEGER_INSERT, 23139,
      INTEGER_INSERT, 7,
      INTEGER_INSERT, 3,
      INTEGER_INSERT, 6,
      INTEGER_INSERT, 27487)
```

   In the API call, the value of *HIM Status Register = %6x* would still correspond to the same *INTEGER_INSERT, 27487* line, and *T/R Status Register = %2x* would still correspond to the same *INTEGER_INSERT, 23139* line.

5. **Message 197 Incorrectly Utilizing Non-Positional Inserts:**
   Gateway %1s error signal from HIM during command issue.
   T/R Status Register = %x HIM %3o CARD %4o FUNC %5o
   HIM Status Register = %6x.

   Note: Non-Positional Inserts must either all be explicitly specified, or not used at all. In this case, the *insert T/R Status Register = %x* is not using a Non-Positional Insert, but the others are.

6. Hypothetical Message Using Literal Percent Sign:
Local Disk is Currently %d%% Full.

Note: To specify a literal percent symbol, type two percent symbols together ("%%"). This will yield the result:

"*Local Disk is Currently 90% Full.*"

Where *90* is the value of integer insert *%d*.

3.2.4 Recorded Data

N/A.

3.2.5 System Message Catalog Printer Formats

N/A.

3.2.6 Interprocess Communications (C-to-C Communications)

N/A.

3.2.7 System Message Catalog External Interface Calls (API Call Formats)

N/A.

3.2.8 System Message Catalog Table Formats

The Message Catalog database fields are described below.

Field	*Data Type (Size)*	*DBM Name*	*Comment*
"Key"	Message Mnemonic	Key	Unique key within individual DBM file
CSCI	Char (3)	csci	pull-down menu of all CSCIs
Author Email	Char (30)	email	
Author Last Name	Char (30)	lastname	
Author First Name	Char (15)	firstname	
CSC	Char (8)	csc	pull-down menu of all CSCs
Message Mnemonic	Char (65)	mnemonic	Must all be in uppercase and valid chars
Number Inserts	Integer (2)	numberinserts	Maximum value of 30
Message Text	Char (200)	messagetext	
Severity Level	Char (1)	severity	"W", "E", or "I"
Alarm	Integer (1)	alarm	0 = Off, 1 = On
Help Text	Char (4000)	helptext	
Release	Char (12)	release	Currently, a pull-down of the CLCS releases.
SPR/ESR Number	Char (12)	esr_spr	SPR or ESR Number. Not a required field.
RSYS	Char (8)	rsys	pull-down menu of all RSYSs
Message Type	Integer (1)	messagetype	Integer value of defined Message Type.
Insert Details Array	Char (270)	detailarray	Comma Delimited list of Detail Numbers
Message Number	Integer (8)	number	Number internally generated

Figure 17. SCID/TCID Message Catalog Database Fields

The RSYS, CSC and Message Type fields are abbreviated constants that can resolve to their full string values. A lookup table provided by SMS contains the definitions.

Multiple TCID DBM Catalogs may exist. For purposes of certification, a separate catalog is created for each RSYS. Message Numbers are incremented internally to the respective catalog. Therefore message numbers are qualified by including the RSYS. Note that only one SCID DBM Catalog is ever created.

Insert Details will be defined in a separate DBM file. The fields within this file are described below.

Field	*Data Type (Size)*	*DBM Name*	*Comment*
"Key"	Detail Number	Key	Unique key within individual DBM file
Detail Text	Char (2000)	detailtext	
Detail Number	Integer (8)	detailnumber	Number internally generated

Figure 18. SCID/TCID Insert Details Database Fields

Because Insert Details may be used and reused by any RSYS, there are only ever two Insert Details DBM files, one for SCID and the other for TCID.

3.2.8.1 CSCI Include File

The CSCI Include File will contain the mnemonics in order to compile the C/C++ source code. To avoid unnecessary recompiles, a separate Include File will be created for each CSCI (assuming the CSCI generates 1 or more messages).

For example, it is possible to have 1 include file for the Gateways (cgs_msgs.h):

MNEMONIC	Message_Number
`#define CGS_FD_OUT_OF_LIMITS`	`1`
`#define CGS_INVALID_COMMAND_RECEIVED`	`15`
`#define CGS_INVALID_ROUTING_CODE`	`30`
`#define CGS_ACTIVE_REQUESTING_SWITCHOVER`	`55`

Steps:

- The MNEMONIC is defined by the author upon creation from the Web Message Form GUI screen.
- The Message Number is assigned internally by the Web Message Form (sequentially assigned) upon successful submission of the Message in the database.
- As part of the Web Message Form, the GUI verifies that the MNEMONIC is unique and has not already been used by another Message.
- As part of SCID Build, the smc_create_index program is executed to create the SCID On-Line Message Catalog and respective Include File(s) are created.
- As part of TCID Build, the smc_create_index program is executed to create the TCID On-Line Message Catalog(s) and respective Include File(s) are created.
- The naming convention for the include files will be ***xxx_msgs.h*** where ***xxx*** is the CSCI name (e.g., ***cmd_msgs.h*** for the CMD CSCI)

3.2.8.2 CM Compile/Make

- As part of the SCID Build, the appropriate source files get compiled/linked based on the MAKE INCLUDE file dependencies.
- As part of the TCID Build, the appropriate source files get compiled/linked based on the MAKE INCLUDE file dependencies.

3.2.8.3 On-Line Message Catalog

Please Reference *Figure 19* for a pictorial representation of the On-Line Message Catalog.

The SCID On-Line Message Catalog below is created as part of the SCID Build process, and each TCID On-Line Message Catalog is created as part of the TCID Build process. Each DBM Message Catalog contains all of the message text, help text and associated fields; e.g., severity level, alarm, CSCI. Each On-Line Message Catalog is indexed by Message Number. The (SMS) API must make 2 references to the Indexed File; the first reference is to the beginning part of the indexed file to retrieve the file byte offset to the location of the Message Text. The 2nd reference is to the actual file location of the Message Text to be retrieved.

The Help Text is retrieved as follows: the offset to the Help Text is calculated based on the Message Text offset + Message Text size. This gives the file byte offset to the Help Text.

Insert Details are appended to the end of the On-Line Message Catalog. The Insert Details Offset section consists of an array of 30 positions. Each position contains the relative offset the Insert Detail is from the end of the last Message Help Text entry. Therefore, actual Insert Detail positions are calculated based on the Header Size + Total Msg Size + Insert Offset. This scheme allows details to be reused by multiple messages within the same On-Line Message Catalog. The Total Msg Size value will be retrieved the first time it is needed, and will be stored in memory for subsequent uses by SMS. This keeps the number of disk accesses to the Message Catalog at two, after the initial retrieval.

The CSCI, CSC, and RSYS are null-terminated strings. The actual text is 3, 8, 8 characters long respectively.

The Message Type is a 1 byte integer, and corresponds to one of the following constants:

1 = ”Summary”
2 = ”Intermediate”
3 = ”Details”
4 = ”Other”

Total_Msg_Cnt represents the total allowed number of messages in the Message Catalog. It provides an upper bound for purposes of validating that a message number is within the range of the catalog. This is also done for reuse of deleted message numbers. When a message is deleted, the Message Text and Help Text are removed from the On-Line Message Catalog, but the Header space remains with all fields set to NULL. Therefore, a new message number will occupy the same space held by a previously deleted message.

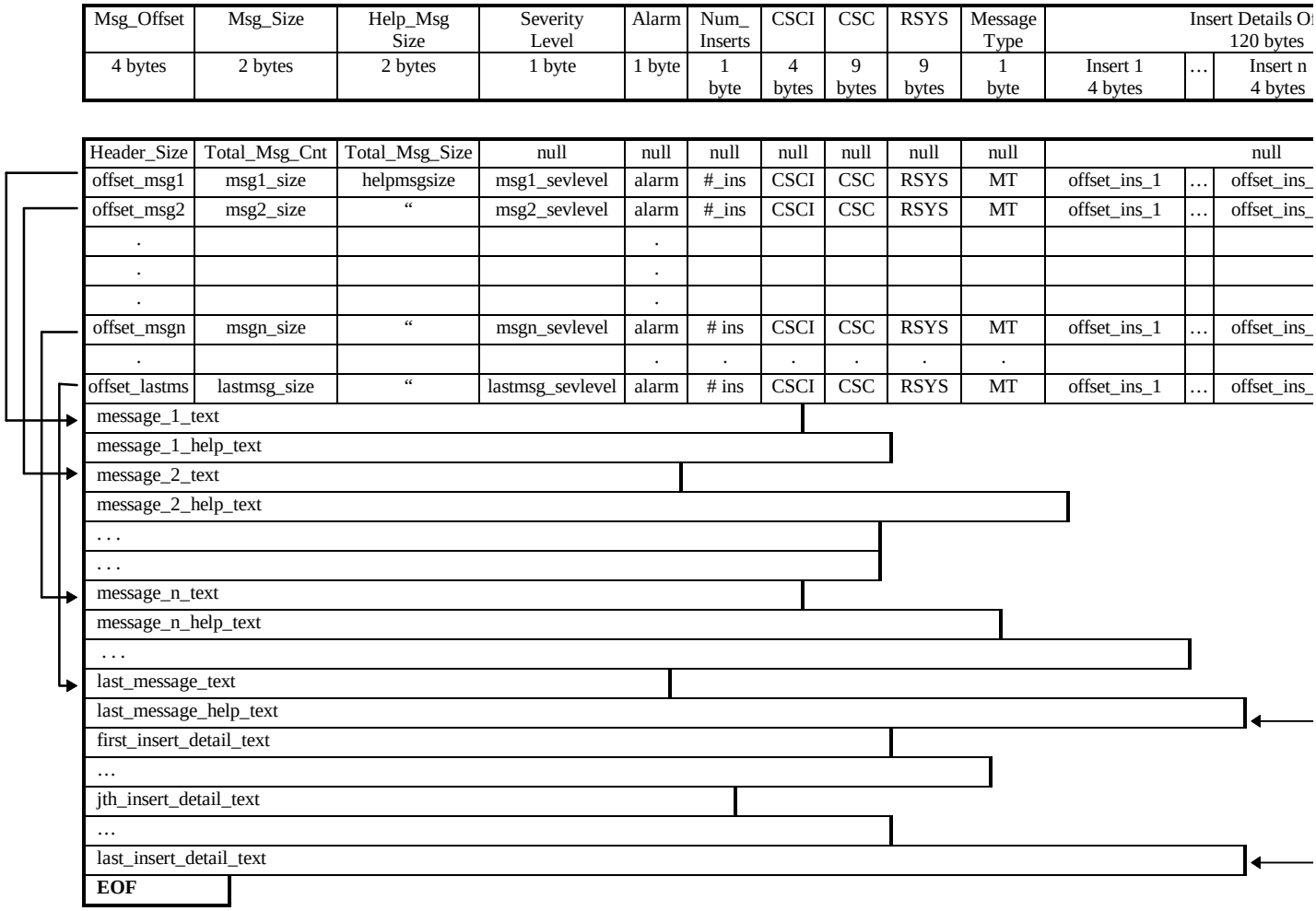

Figure 19. Pictorial representation of the On-Line Message Catalog, "*smc_xxxx_messages.idx*".

Note: The CSCI Insert Types are not required in the Index File.
The file byte offset to the Help Text is not required. Help Text offset is calculated as follows:
Help Text Offset = Message Text Offset + Message Text Size.
Insert Details offsets are relative to the end of the message and help text. Actual positions are calculated as follows:
Insert Details Position = Header Size + Total Message Size + Insert Offset.

3.2.9 System Message Catalog Test Plan

System Message Catalog system-level tests will be executed in the SDC environment as indicated. There are no special hardware configurations required.

The specific test cases that will be run include:

1. Test the password protection by attempting to enter several of the CSCIs.
2. Create several messages in the SCID and TCID Message Catalogs.
3. Execute the smc_create_index program to create the SCID and TCID On-Line Message Catalogs.
4. Verify the correctness of the SCID and TCID On-Line Message Catalogs via a test driver that executes the SMS_GET_MESSAGE API.
5. Delete a message from the each of the SCID and TCID Message Catalogs.
6. Execute the smc_create_index program to create the SCID and TCID On-Line Message Catalogs.
7. Verify the correctness of the SCID and TCID On-Line Message Catalogs via a test driver that executes the SMS_GET_MESSAGE API.
8. Similar to above, retrieve the HELP Text via SMS_GET_MESSAGE_HELP API via a test driver that executes the API.
9. Similar to above, retrieve the Insert Details via SMS_GET_MESSAGE_INSERT_FROM_DATABASE API via a test driver that executes the API.
10. Verify the correctness of the CSCI Message Include Files via inspection and compilation of various CSCI Include files.

Appendix A

The example below shows the message layout in the CCMS System Message Catalog for CCMS System Message #197.

```
MESSAGE NUMBER:  197                MESSAGE TEXT

      MM/SS FEP 197 ($$$$) ERROR SIGNAL FROM HIM DURING COMMAND ISSUE
 T/R STATUS REG = $$$$ HIM $$$ CARD $$ FUNC $
 HIM STATUS REG = $$$$

 MESSAGE COLOR:  CYAN              PREFIX:     FEP
 PRINTED:  SPA PRINTER (SPA)       DISPLAYED:  SYSTEM MSG PAGE (SMP)
 INSERTS:     1) QASC       2) HEX        3) OCTAL      4) OCTAL
 5) OCTAL     6) HEX

 INSERT DESCRIPTION:

 1) GSE FEP NAME
 2) TRANSMITTER/RECEIVER STATUS REGISTER CONTENTS AT TIME
 OF HIM ERROR
 3) HIM NUMBER OF HIM DETECTED TO HAVE HIM ERROR
 4) CARD NUMBER OF HIM DETECTED TO HAVE HIM ERROR
 5) FUNC CODE OF HIM DETECTED TO HAVE HIM ERROR
 6) CONTENTS OF HIM STATUS REGISTER
 BIT 15 = N/A
 BIT 14 = N/A
 BIT 13 = NACK FROM I/O CARD
 BIT 12 = MULTIPLE ACK
 BIT 11 = READY TIMEOUT FROM I/O CARD
 BIT 10 = N/A
 BIT 9  = LOGIC ERROR
 BIT 8  = POWER SUPPLY OVER TEMP


 DURING COMMAND ISSUANCE ONE OR MORE OF THE FOLLOWING NON-TERMINAL
 ERRORS OCCURRED: ACKNOWLEDGE TIMEOUT, MULTIPLE ACKNOWLEDGE,
 READY TIME OUT, LOGIC ERROR, OVER TEMPERATURE WARNING

 ACTION TAKEN:

 HIM STATUS REGISTER RESET ISSUED FOR ALL ERRORS, EXCEPT OVER
 TEMPERATURE WARNING. STATUS REG 1 ERROR IS REPORTED FOR
 HARDWARE MONITOR INFORMATION. NOTE: CONSECUTIVE OVER TEMP
 WARNINGS ARE NOT REPORTED.

 USER OPTIONS:

 NOTIFY HARDWARE ENGINEERING

 SPR/CR      RELEASE ID       DESCRIPTION
 K3.0387     F912             ENHANCEMENTS TO HIM ERROR MESSAGES
```

End